An Optimal Smoothing Approach for Trajectory Reconstruction in Planetary Exploration

Amit Brandes

Department of Computer and Systems Science
University of Rome, 'La Sapienza'
brandes@dis.uniroma1.it

Keywords: localization; planetary exploration; optimal smoothing; nonlinear optimization; mobile robot.

Abstract

This paper deals with fusion of real-time and off-line measurements, formulated in terms of a nonlinear least-squares optimization problem, and solved analytically by linearizing the measurement model. The proposed approach is applied to a planetary exploration rover, equipped with an EKF based localization system using fixations to unknown fixed landmarks. The smoothing method enables fusing the off-line measurements, related to the landmarks, with the real-time EKF estimates. The analytic solution of the resulting optimization problem has major advantages over standard methods, avoiding risks of solution divergence and of convergence to local minima, and reducing the computational load. An analytic evaluation of the linearization error is presented, along with simulation results to demonstrate the effectiveness of the proposed approach.

1. Introduction

During recent years the localization of planetary rovers has been an important research field within the large theme of planetary exploration, see for example [9,10,14]. The localization algorithms serve two major aspects. The first is to enable reasonable estimation of the rover position and orientation (pose) that will allow reaching pre-designed goals. The second aspect is the mapping of the environment to help future localization and to serve further research on future missions. When distinguishing between the above tasks the requirements from each task can be viewed and defined separately. For the former real-time operation is crucial, whereas accuracy can be limited to some range of operability restrictions. Such limits can be, for example, the range of on-board cameras that command the last phase of a trajectory toward a pre-designed goal during an exploratory mission. For the mapping task, on the other hand, it is desired to maximize accuracy, even at the expense of non real-time processing. Measurements that arrive off-line can hardly help resolving the localization task but can contribute to the solution of the mapping task.

The MARS (Mobile Autonomous Robotic System) project is aimed at studying and developing techniques and methods for space and planetary exploration. It is part of a series of projects relating with the control of space systems [4,5]. Within the framework of this project a mobile platform, to be implemented as an exploratory rover, was developed. The MARS rover is equipped with a camera system enabling fixation on points of interest during motion, and producing on-line range and direction measurements to the fixation points. The rover is also equipped with wheel encoders and will have some other device for absolute orientation measurement, such as a sun sensor. A real-time extended Kalman estimator for the localization of the MARS rover was developed, in which estimation is done by fusing range and angle measurements to fixed unknown landmarks, along with the rover proprioceptive sensors' data [2]. This system provides very accurate velocity estimates and good localization in a relative sense, but fails to remove initial position errors.

In this paper an optimal smoothing approach is introduced, enabling trajectory reconstruction and correction of localization errors by fusing off-line measurements of the scene with the real-time estimates. The problem of fusing real-time and off-line measurements had been widely studied by several authors. When the underlying models are linear it is very convenient to use Kalman smoothers, e.g. [1,7,13], though it requires a forward/backward filtering strategy that implies at least two processes of recursive computation over the whole trajectory. If, on the other hand, the system is strongly non-linear, then it is customary either to formulate a linear cost function subject to nonlinear constrains as in [11], or to formulate a non-linear cost function and minimize it by some standard non-linear minimization method [12,15]. Such methods suffer from the risks of instability and of convergence to local minima. Furthermore, the iterative procedures used to solve the minimization problem might even diverge, and are quite "expensive" in terms of computer resources.

Following [12] and [15], data fusion is reformulated as a nonlinear least square optimization problem. A cost function with a non-linear measurement model is obtained, which can be solved for a minimizing solution by applying some

standard method, e.g. a quasi-Newton optimization process. To avoid the risks associated with iterative non-linear optimization methods, it is also proposed to use a linearized measurement model and compute a closed form solution. Using this approximation, the disadvantages of standard methods are avoided and the computational load is highly reduced. The approximation error is then analyzed to show the validity of the proposed linearized method to the underlying scene of the case study.

The remainder of this paper is organized as follows: In the next section the proposed approach is presented followed by a presentation of the linearization method enabling an analytic solution. Then, in section 4 the approximation is analyzed and validated for the underlying scene. A simulation study confronting the proposed method with a standard non-linear optimization method is presented in section 5.

2. Optimal least-squares estimation approach

Consider a non-linear dynamic system of the following form:

$$\begin{aligned} x_{k+1} &= f\left(x_k, u_k, \eta_k\right) \\ y_k^{(1)} &= h^{(1)}\left(x_k, \pi_k^{(1)}\right) \end{aligned} \tag{1}$$

where the real-time measurement $y^{(1)}$ is used to construct an on-line state observer producing real-time estimates of the system state:

$$\hat{x}_k = g\left(u_k, y_k^{(1)}\right) \tag{2}$$

Another measurement, $y^{(2)}$, is available, though it is obtained with significant latency and cannot be used on-line:

$$y_j^{(2)} = h^{(2)}\left(x_j, \pi_j^{(2)}\right) \tag{3}$$

$\eta, \pi^{(1)}, \pi^{(2)}$ are assumed to be Gaussian, uncorrelated additive white noises. $y^{(2)}$ might be asynchronous with $y^{(1)}$, but the *j* time instants are well defined by the *k* instants, i.e. if $y_{\bar{j}}^{(2)}$ is defined at some time $\bar{t}$, then there exists $\bar{k}$ such that $\bar{t} = \bar{k} \cdot T$, where *T* is the system period. This work focuses on improving the state estimate by using the off-line measurement $y^{(2)}$.

Referring to the MARS case, a filtered state estimate is available, based on real-time measurements of distance and angle to some fixed unknown landmarks. The locations of the landmarks used (in global coordinates) are unknown when the on-line localization is performed. They can be derived, however, off-line from imagery systems of the lander and the rover. Once this information is available it is required to compute an improved (smoothed) state estimate ($\hat{\hat{x}}$), based on the filtered estimate ($\hat{x}$).

In what follows it is assumed that for a certain time instant $\bar{k}$ there exists an initial estimate $\hat{x}_{\bar{k}}$ and its estimation error covariance matrix. Furthermore, a measurement $y_{\bar{j}}^{(2)}$ true for the same time instant is available such that:

$$y_{\bar{j}}^{(2)} = h^{(2)}\left(x_{\bar{k}}, \pi_{\bar{k}}^{(2)}\right) \tag{4}$$

Omitting the time indices for simplicity, the following cost function is constructed at each time instant:

$$\begin{aligned} J = &\left\langle P_x^{-1}(\hat{x} - x)\,,\,(\hat{x} - x)\right\rangle + \\ &+ \left\langle R^{-1}(y^{(2)} - h^{(2)}(x,0))\,,\,(y^{(2)} - h^{(2)}(x,0))\right\rangle \end{aligned} \tag{5}$$

The brackets indicate the inner product operator. P_x and R are positive definite weight matrices that can be interpreted as follows: a) The first term of J accounts for the "price" of changing the first estimate. A natural choice for its weight matrix is the inverse of the covariance matrix (accuracy) of the first estimation error. b) The second term accounts for the difference between the predicted measurement and the actual measured value. This term is weighed by the inverse of the measurement accuracy so that *R* is the covariance matrix of the measurement noise $\pi^{(2)}$. Using this setting, the smoothed state is the minimizing solution of the cost function *J*:

$$\hat{\hat{x}} = \arg\min_x J \tag{6}$$

Equation (6) is solvable by applying one of the many existing non-linear optimization procedures. An application of one such a method to the MARS case, using MATLAB built-in routines, is demonstrated in section 5. The results are very accurate but the algorithm suffers from the usual problems of non-linear optimization methods: the need for iterative solution, the risk of convergence to local minima, and the relatively high computational load.

It should be observed at this point that if a linear measurement model is assumed in equation (4), then equation (5) becomes quadratic in the variable *x*. The problem is then strictly convex, and an analytic solution can be obtained [6]:

$$\hat{\hat{x}} = \left(P_x^{-1} + H^T R^{-1} H\right)^{-1} \cdot \left(P_x^{-1}\hat{x} + H^T R^{-1} y^{(2)}\right) \tag{7}$$

where *H* is the linear measurement model, i.e.

$$h^{(2)}(x,0) = H \cdot x \tag{8}$$

The first term in equation (7) can be interpreted as the smoothing error covariance matrix:

$$\mathrm{cov}(x - \hat{\hat{x}}) = \left(P_x^{-1} + H^T R^{-1} H\right)^{-1} \tag{9}$$

3. Linearization of the measurement model

Planetary exploration missions present severe limitations in terms of computational load and the reliability of the solutions is of extremely high importance. Those constrains make it easy to understand why the closed-form formula of equation (7) have large benefits over other possible procedures to solve equation (6). By introducing a nonlinear coordinate change, the problem can be reformulated as a quadratic optimization problem and solved analytically by means of equation (7).

Let $z = l(x)$ be a coordinate change which leads to a linear measurement model:

$$y^{(2)} = H \cdot z + \pi^{(2)} = H \cdot l(x) + \pi^{(2)} \tag{10}$$

The unknown z can be now estimated by applying equation (7) and substituting $\hat{x}$ with $\hat{z} = l(\hat{x})$. No modification is required for the R matrix since:

$$\begin{aligned} R &= \mathrm{cov}(\pi^{(2)}) = \mathrm{cov}(y^{(2)} - h^{(2)}(x,0)) = \\ &= \mathrm{cov}(y^{(2)} - H \cdot l(x)) \end{aligned} \tag{11}$$

Unfortunately this approach is not valid for the P matrix since $\mathrm{cov}(x - \hat{x}) \neq \mathrm{cov}(l(x) - l(\hat{x}))$. Next an approximation for the latter covariance matrix, regarding the planetary rover case, is computed and evaluated. Using this approximation the optimization problem can be solved analytically by applying:

$$\hat{\hat{z}} = \left(P_z^{-1} + H^T R^{-1} H\right)^{-1} \cdot \left(P_z^{-1} \hat{z} + H^T R^{-1} y^{(2)}\right) \tag{12}$$

where P_z is the approximated covariance matrix, i.e. $P_z \approx \mathrm{cov}(z - \hat{z}) = \mathrm{cov}(l(x) - l(\hat{x}))$.

Regarding now the special case of the MARS rover, the state vector x consists of the rover pose - (x_r, y_r, θ_r) and the camera measurements (ρ, ψ) - range and angle to the observed landmark. See also Figure 1 and [2] for further details.

$$x = (x_r, y_r, \theta_r, \rho, \psi)^T \tag{13}$$

The non-linear measurement $y^{(2)}$ is the landmark location in global coordinates:

$$y^{(2)} = (x_{LM}, y_{LM})^T + \pi^{(2)} \tag{14}$$

where $\pi^{(2)}$ is the measurement noise. The exact formulation of measurement function is as follows:

$$h^{(2)}(x, \pi^{(2)}) = \begin{bmatrix} x_r + \rho \cdot \cos(\theta_r + \psi) \\ y_r + \rho \cdot \sin(\theta_r + \psi) \end{bmatrix} + \pi^{(2)} \tag{15}$$

A coordinate change which serves to linearize the measurement model is:

$$z = (x_r, y_r, \rho \cdot \cos(\theta_r + \psi), \rho \cdot \sin(\theta_r + \psi))^T \equiv l(x) \tag{16}$$

One then obtains:

$$y^{(2)} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \cdot z + \pi^{(2)} \equiv Hz + \pi^{(2)} \tag{17}$$

For the application of the smoother, the covariance matrix of the transformed vector is needed. The latter can be computed with the help of a linear approximation of the coordinate change *l(x)*:

$$P_z \cong \left.\frac{\partial l}{\partial x}\right|_{\hat{x}} \cdot P_x \cdot \left.\frac{\partial l}{\partial x}\right|_{\hat{x}}^T \tag{18}$$

where $\hat{x}$ is the estimated state from the EKF and P_x its estimated error covariance matrix. Smoothing can now be performed by applying equation (12).

4. Validation of the linear approximation

In this section the linear approximation is justified by evaluating the approximation error for the relevant working environment. The following vectors are now considered:

$$\begin{aligned} s &= [x \quad y \quad r \quad \alpha]^T \\ t &= [x \quad y \quad r \cdot \cos(\alpha) \quad r \cdot \sin(\alpha)]^T \equiv l(s) \end{aligned} \tag{19}$$

Let $\hat{s}$ be an estimate of s, with P_s the estimation error covariance matrix. Define now the following covariance matrices:

$$\begin{aligned} P_t &= \mathrm{cov}(l(s) - l(\hat{s})) && \text{(a)} \\ P_{lin} &= \hat{L} \cdot P \cdot \hat{L}^T \approx P_t && \text{(b)} \end{aligned} \tag{20}$$

where $\hat{L}$ is the Jacobian matrix of the coordinate change, calculated at the estimate $\hat{s}$. The linear approximation can be evaluated by assessing the difference $\|\Delta P\| = \|P_t - P_{lin}\|$, where it is convenient to use the trace of the matrix as its norm, i.e. $\|P\| = trace(P)$.

In what follows $\hat{r}$ is the estimated range, contrary to r which is the true range, and similarly for the angle α.

Taking $P_s = \begin{bmatrix} p_x & <> & & \\ & p_y & & <> \\ <> & & p_r & \\ & <> & & p_\alpha \end{bmatrix}$, with arbitrary off-diagonal elements, one can easily compute:

$$\|P_{lin}\| = \left\|\hat{L} \cdot P_s \cdot \hat{L}^T\right\| = p_x + p_y + p_r + \hat{r}^2 \cdot p_\alpha \tag{21}$$

whereas for the true distribution the following can be demonstrated [8]:

Given that $r - \hat{r}$ and $\alpha - \hat{\alpha}$ have zero mean Gaussian distribution, i.e.

$$\begin{aligned} r - \hat{r} &\approx N(0, \sqrt{p_r}) && \text{(a)} \\ \alpha - \hat{\alpha} &\approx N(0, \sqrt{p_\alpha}) && \text{(b)} \end{aligned} \tag{22}$$

it follows that:

$$\begin{aligned} &\mathrm{var}(r \cdot \cos\alpha - \hat{r} \cdot \cos\hat{\alpha} \mid r, \alpha) = \\ &= p_r e^{-p_\alpha} [\cos^2\alpha \cosh(p_\alpha) + \sin^2\alpha \sinh(p_\alpha)] + \\ &+ r^2 e^{-p_\alpha} [\cos^2\alpha(\cosh(p_\alpha) - 1) + \sin^2\alpha \sinh(p_\alpha)] \end{aligned} \quad \text{(a)}$$

$$\begin{aligned} &\mathrm{var}(r \cdot \sin\alpha - \hat{r} \cdot \sin\hat{\alpha} \mid r, \alpha) = \\ &= p_r e^{-p_\alpha} [\sin^2\alpha \cosh(p_\alpha) + \cos^2\alpha \sinh(p_\alpha)] + \\ &+ r^2 e^{-p_\alpha} [\sin^2\alpha(\cosh(p_\alpha) - 1) + \cos^2\alpha \sinh(p_\alpha)] \end{aligned} \quad \text{(b)} \tag{23}$$

Since the x,y members of s are not affected by the coordinate change l, direct computation will show that:

$$\|P_t\| = p_x + p_y + p_r + r^2 \cdot (1 - e^{-p_\alpha}) \tag{24}$$

An upper bound for the approximation error can be assessed now by computing:

$$\begin{aligned}\|\Delta P\| &= \|P_t - P_{lin}\| \le \|P_t\| - \|P_{lin}\| = \\ &= r^2(1-e^{-p_\alpha}) - \hat{r}^2 p_\alpha = \\ &= r^2(p_\alpha - {p_\alpha^2}/{2} + {p_\alpha^3}/{6} - \cdots) - \hat{r}^2 p_\alpha \\ &= (r^2 - \hat{r}^2)p_\alpha + r^2(-{p_\alpha^2}/{2} + {p_\alpha^3}/{6} - \cdots)\end{aligned} \tag{25}$$

When navigating the MARS rover, a reasonable requirement from the real-time localization system is a maximal error of $O(10^0)$ meters in position, and of $O(10^{-2})$ radians (=$O(10^0)$ degrees) in orientation. The range estimation error is smaller than the range measurement error, which is of $O(10^0)$ meters, while a reasonable range to landmarks is of $O(10^1)$ meters. Bearing in mind that $\alpha = \theta_r + \psi$, and that the direction to the landmark, ψ, is much more accurate that the orientation θ_r, it follows that the accuracy of α is within the same order of magnitude as that of the orientation. Taking therefore:

$$\begin{aligned}&p_x, p_y \approx O(10^1) \\ &p_r \approx O(10^0),\ r \approx O(10^1),\ dr \approx O(10^0) \\ &p_\alpha \approx O(10^{-4}),\end{aligned} \tag{26}$$

where *dr* is the range estimation error, equation (25) can be further simplified by neglecting the last term with respect to the first one and using the fact that *dr* is small with respect to *r*:

$$\|\Delta P\| \le (r^2 - \hat{r}^2)p_\alpha \approx 2r \cdot dr \cdot p_\alpha \tag{27}$$

Finally substituting the values in (26) into equation (27) it can be computed that:

$$\frac{\|\Delta P\|}{\|P_t\|} \approx O(10^{-3}) \tag{28}$$

hence demonstrating that deviations from the true value are of negligible order which validates the use of the linear approximation. Further justification for the use of the proposed approximate method is obtained by a simulation study presented next.

5. Algorithm evaluation – a simulation study

A return mission trajectory, passing through several points of interest and covering roughly 250 meters, was simulated in MATLAB. The main error source is slippage which is a major problem in unstructured outdoors environment. Other error sources are sensors measurement errors, which are simulated as random noises.

A first estimate of the rover position is calculated in real-time using the EKF estimator of [2]. The estimator makes use of on-line measurements available from a camera system that produces range and direction measurements to fixed landmarks. Two different landmarks are used. For the first part of the trajectory, measurements of an arbitrary landmark located at (100,60) are taken. Then, for the return journey, the lander located at the origin serves as a landmark. The location of the landmarks is unknown at the time of calculation, but the fact the they are fixed is implicit in the filter equations and enables very good estimation of true velocities.

As presented in [2], the EKF estimator is not capable of removing initial position errors. Furthermore, the position estimation error covariance is growing unboundedly. A Monte-Carlo simulation of the EKF estimator, consisting of 50 different runs showed that the EKF produces a position estimation error that varies between 3 to 6 meters, with an estimation error covariance starting at 25 m^2 (initial conditions) and reaching 170 m^2 (13 m 1σ error).

The EKF estimate is first smoothed using a classic optimization method. Optimization is based on standard MATLAB routines executing a quasi-Newton method that uses the BFGS (Broyden-Fletcher-Goldfarb-Shanno) formula to predict the Hessian matrix of the cost function. After determining the direction of search, the minimizing solution is obtained by implementing a cubic-polynomial line search [3]. A second estimate is calculated using the closed form solution presented above. The two methods are compared in terms of accuracy, number of iterations and computation time. The graphs shown are based on a 50 run Monte-Carlo simulation. For the first landmark, a position error of 1 meter 1σ on each direction (*x* and *y*) is used, having $R = diag([1,1])$. For the return trajectory a better accuracy of 0.5 meter 1σ, $R = diag([0.25, 0.25])$.

5.1 Simulation results

Figure 2 presents the whole simulated trajectory, clearly showing an improvement in the position estimation. The EKF estimate is biased from the true position due to initial condition errors that are not removed. The two smoothed estimates remove this bias, leaving only a small estimation error. The difference between the two estimates (centimeters) is too small to be distinguished on this scale. Figure 3 further illustrate the same phenomena, as the initial condition errors in the EKF remain throughout the whole trajectory, but are then removed by the smoothing process. Note in Figure 4 that the 1σ values for the two smoothing methods are clearly bounded and that they are closely related to the accuracy of the landmark measurements (the *R* matrix). Note also the clear effect of the landmark change after 250 seconds of motion. The average difference between the two smoothed estimates is within a few centimeters order of magnitude (Figure 5), whereas the standard deviation of both is within tens of centimeters order of magnitude, demonstrating no practical difference in accuracy between the two methods.

Another important issue of comparison is the computational load required by each of the two smoothing processes. It was chosen to test this issue in a relative manner, by comparing average execution time per estimated point for each method when executed on the same processor. The execution time on a specific processor has no important significance as its own,

but the comparison can shed light on the relative efficiency of the two processes. Using an AMD Athalon XP1700 processor with 252 MBRAM, an average of 40 msec is needed to compute a smoothed estimate for each point of interest (Figure 6 down), with 10-15 iterations of the minimization procedure (Figure 6 up). The approximated method, on the other hand, requires an average of 0.7 milliseconds to compute a smoothed estimate per each point of interest, executing only one single calculation per point, thus demonstrating circa 50 times higher efficiency.

6. Conclusions

A smoothing process based on optimal estimation approach was here presented, intended to fuse off-line measurements and to improve the accuracy of real-time state estimates of a dynamic system. By introducing a nonlinear coordinate change the optimization problem is solved analytically avoiding the risks of non-linear optimization methods. In particular this approach is implemented to improve the accuracy of a localization process for a planetary rover and its validity demonstrated for the underlying scene. The higher efficiency, and moreover the advantage of the closed-form solution, are of very high importance when dealing with planetary exploration, since problems related with iterative minimization procedure (e.g. possible divergence of the solution, convergence to local minima, the possibility of not finding a solution within a small enough number of iteration) can highly reduce the autonomy of a planetary rover. Hence the ability to reach a similar accuracy while avoiding such problems is of high benefit for a planetary rover, as is the significantly higher efficiency of the proposed method.

It should also be pointed out that the same approach can be followed for any system that comply with a similar structure, i.e. some off-line measurements available with significant delay. If some initial estimate can be computed, it can be used as an initial guess for a constrained least squares optimization similar to the one presented here. To simplify the solution and to lower computational load, non-linear measurement models can be linearized by using an adequate state transformation, and the linearization validated following the guidelines of this paper.

Acknowledgements: The research is conducted at the Systems and Control Laboratory, Department of Computer and Systems Science (DIS), University of Rome "La Sapienza", headed by prof. S. Monaco, and is funded by the Italian Space Agency (ASI).

References

[1] Betaille, D., and Bonnifait, P., "Road Maintenance Vehicles Location Using DGPS, Map-Matching and Dead-Reckoning: Experimental Results of a Smoothed EKF", *Proceedings of the ION Annual Meeting*, San Diego, CA, (2000).

[2] Brandes, A., "Filter/Smoother Localization for Outdoors Applications", *Proceedings of the 11th International Conference on Advanced Robotics*, Coimbra, Portugal, (2003).

[3] Coleman T. *et al.*, *"Optimization Toolbox User's Guide"*, The Mathworks Inc., Natick, MA, (1999).

[4] Di Gennaro S., Monaco S. and Normand-Cyrot D. "Nonlinear Digital Scheme for Attitude Tracking", *Journal of Guidance Control and Dynamics*, **vol. 22**, no. 3, pp. 467-478, (1999).

[5] Di Giamberardino P., Monaco S. and Ronchini R., "Nonlinear Regulation for Low Orbit Tracking of a Spacecraft for a Scientific Mission", *Proceedings of the International Symposium on Space Dynamics*, CNES, (2000).

[6] Gelb, A., (Editor), *"Applied Optimal Estimation",* MIT press, Cambridge, MA, (1974).

[7] Hagen, S., and Krose, B., "Trajectory Reconstruction for Self-Localization and Map Building", *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA)*, **Vol. 2**, pp. 1796-1801, (2002).

[8] Lerro, D., and Bar-Shalom, Y., "Tracking with Debiased Consistent Converted Measurements Versus EKF", *IEEE Transactions on Aerospace and Electronic Systems*, **Vol. 29**, No. 3, pp. 1015-1022, (1993).

[9] Matijevic, J., and Shirley, D., "The Mission and Operation of the Mars Pathfinder Microrover", *Journal of Control Engineering and Practice*, **Vol. 5**, No. 6, pp. 827-835, (1997).

[10] Olson, C.F., *et al.*, "Robust Stereo Ego-Motion for Long Distance Navigation", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, **Vol. 2**, pp. 453-458, (2000).

[11] Psiaki, M. L., *et al.*, "Attitude Estimation for a Flexible Spacecraft in an Unstable Spin", *Journal of Guidance Control and Dynamics*, **Vol. 25**, No. 1, pp. 88-95, (2002).

[12] Psiaki, M. L., "Autonomous LEO Orbit Determination from Magnetometer and Sun Sensor Data", *Journal of Guidance Control and Dynamics*, **vol. 22**, No. 2, pp. 305-312, (1999).

[13] Roumeliotis, S. I., *et al.*, "Smoother based 3D Attitude Estimation for Mobile Robot Localization", *Proceedings of the 1999 IEEE International Conference on Robotics and Automation (ICRA)*, **Vol. 3**, pp. 923-931, (1999).

[14] Volpe, R., *et al.*, "Enhanced Mars Rover Navigation Techniques", *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA)*, **Vol. 1**, pp. 923-931, (2000).

[15] Weiss, A.J., and Friedlander, B., "DOA and Steering Vector Estimation Using a Partially Calibrated Array", *IEEE Transactions on Aerospace and Electronic Systems*, **Vol. 32**, No. 3, pp. 1047-1057, (1999).

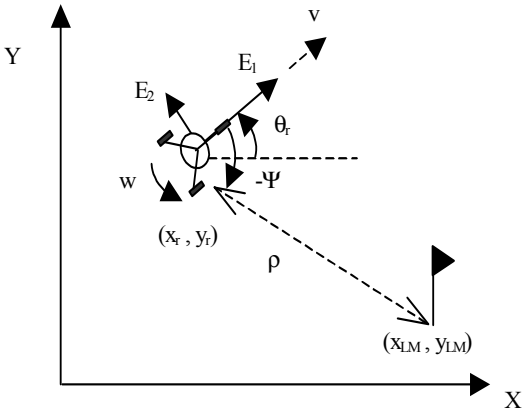

Figure 1: Definition of problem variables.

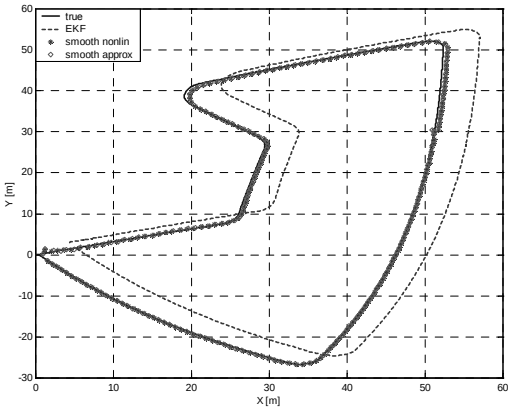

Figure 2: The true trajectory, the EKF estimate, and the estimates by the two smoothing algorithms. The last two almost coincide.

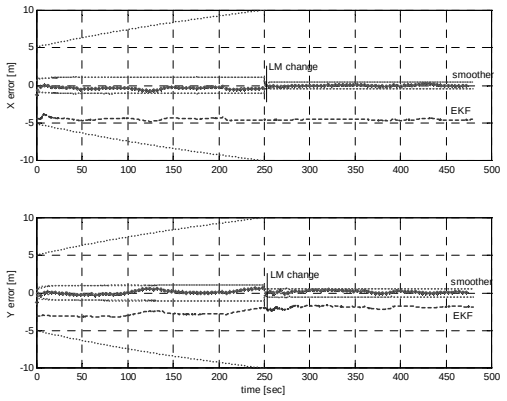

Figure 3 : X-position error (up figure) and Y-position error (down figure) of the EKF and the two smoothing processes. 1-sigma bounds are presented by dotted lines.

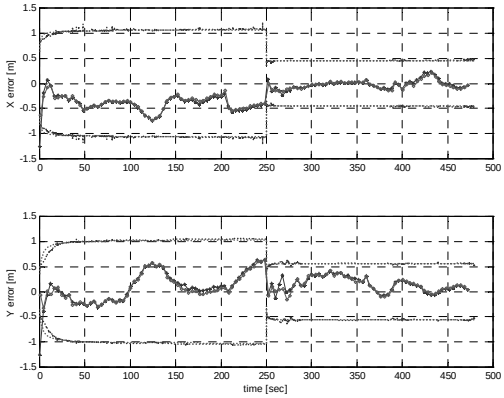

Figure 4: The estimation error of the two smoothing algorithms. 1-sigma bounds are dotted.

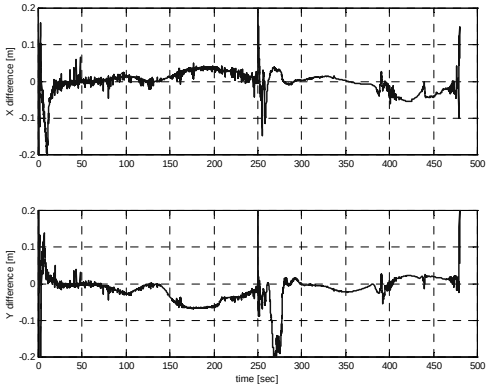

Figure 5: The difference between the two smoothing algorithms.

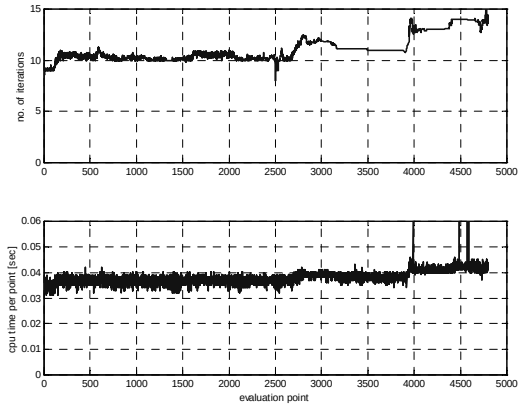

Figure 6: The average number of iterations per smoothed point (up) and the average evaluation time per point (down) for the quasi-Newton optimization method. The average evaluation time for the linearized method is 0.7 msec.